Synchronizing Read & write pointers

(3)

L47: Entry 4 of 20 File: USPT Jul 17, 2001

DOCUMENT-IDENTIFIER: US 6263410 B1
TITLE: Apparatus and method for asynchronous dual port FIFO

Abstract Text (1):
An apparatus and method for controlling an asynchronous dual port FIFO memory is provided. The asynchronous FIFO may operate at frequencies satisfying 0.5f.sub.2 <f.sub.1 <f.sub.2 or 0.5f.sub.1 <f.sub.2 <f.sub.1, where f.sub.2 is the write frequency if f.sub.1 is the read frequency, or vice versa. A FIFO in accordance with the present invention comprises a dual port random access memory, a read pointer, a write pointer, a synchronization circuit and a status indicator. In the FIFO design, the read pointer indicating the read address is a simple sequential counter, and the write pointer indicating the write address is a Gray code counter. Gray code to sequential count converters are used to convert the Gray codes to sequential counts. The synchronization circuit synchronizes the write pointer and the read pointer using a read clock. A status indicator with simple circuits is provided to indicate if the FIFO is almost full or empty.

Brief Summary Text (6):
Basically, the problem of controlling an asynchronous FIFO is that in an asynchronous FIFO, different access frequencies may result in uncertainty of addresses specified by the read pointer and the write pointer. It is thus hard to determine if the current FIFO status is full or empty.

Brief Summary Text (7):
FIG. 2 illustrates an asynchronous dual port FIFO in which the Gray code method is used to design the FIFO. The architecture of FIG. 2 represents one of the most common approaches to solving the problem associated with the unstable memory addresses. This structure reduces the number of bits of an unstable transient state in a read pointer or write pointer to the minimum, while the pointers are being sampled.

Brief Summary Text (8):
In the design of FIG. 2, the asynchronous FIFO comprises two Gray code counters. One is used as a read pointer, and the other is used as a write pointer. To determine how much memory space in the FIFO memory can be accessed, the Gray codes corresponding to the read and write pointers are first converted to sequential counts. A subtraction is then performed on the two sequential counts in order to determine the available space in the FIFO.

Brief Summary Text (9):
However, the design in FIG. 2 has some disadvantages. Because of different access frequencies to an asynchronous FIFO, the relative positions between these two pointers may not actually tell the true use level of the FIFO even with a synchronized circuit implementation. It also requires two status indicator circuits to determine whether the current FIFO status is full or empty.

Brief Summary Text (12):
According to the present invention, a simple sequential counter instead of a Gray code counter is used as the read pointer in an asynchronous dual port FIFO, and the write pointer is a Gray code counter. The write pointer is sampled by a read clock

so as to be synchronized with the read pointer. The read pointer is not synchronized to a write clock.

Brief Summary Text (13):
Using a Gray code counter for the write pointer reduces the number of bits in a transient state to a minimum because in such a counter only one bit changes between two adjacent FIFO addresses. The encoding stability is increased. The read pointer is implemented by a typical sequential counter because the encoding circuit of a sequential counter is better than that of a Gray code counter in terms of timing slacks and circuit areas. In addition, a transient state does not occur in the read pointer because it is not synchronized to the write clock.

Brief Summary Text (14):
To determine how much memory space is available for access, the Gray code corresponding to the write pointer is first converted to sequential format by means of a conversion circuit. After performing subtraction on the two sequential counts of write and read pointers, a status indicator comprising two comparators and a 2's complement block outputs an empty signal if the subtraction result is equal to zero.

Drawing Description Text (7):
FIG. 6 illustrates how the transient state from one address to its next address in the write pointer is sampled by a read clock.

Drawing Description Text (9):
FIGS. 8a-8c illustrate three different possible situations when the write pointer is sampled by a read clock if the frequency of the write pointer is faster than that of the read clock in a general asynchronous dual port FIFO.

Detailed Description Text (3):
A write pointer 304 is used to indicate a write address and a read pointer 305 is used to indicate a read address. Data can be simultaneously written into and read from the RAM 301. The comparison between the write pointer and the read pointer for determining the available memory space in the FIFO also occurs at the same time. The asynchronous dual-port FIFO thus works very efficiently.

Detailed Description Text (4):
The write pointer 304 is implemented by a Gray code counter. The memory address indicated in such a counter only changes one bit between two adjacent FIFO addresses. This reduces the number of bits in an unstable transient state to a minimum when a FIFO address is changed as well as provides increased encoding stability. A write enable signal WE and a write clock WCLK trigger the Gray code counter. When WE is enabled, the Gray Code counter changes one bit at a time following the rising edge transition of WCLK, and the corresponding write pointer 304 points to the next write address.

Detailed Description Text (5):
The read pointer 305 is implemented by a conventional sequential counter. A read enable signal RE and a read clock RCLK trigger this sequential counter. When RE is enabled, the sequential counter is incremented by 1 following the rising edge transition of RCLK, and the corresponding read pointer 304 points to the next read address. In such a design, because the read pointer 305 is implemented by a sequential counter instead of a Gray code counter, a Gray code encoding is not necessary. In practice, the encoding circuit of the sequential counter has better timing slack and requires smaller area.

Detailed Description Text (6):
Because the encoding method of the read pointer differs from that of the write pointer, Gray code to sequential count converters are used to ensure that the dual port RAM addresses in the two pointers are interpreted in an identical manner.

Before the write pointer 304 is sent to the dual port RAM 301, the Gray code is first converted to a sequential count by the a first Gray code to sequential count converter 306. The output of a synchronizing to read clock circuit 307 that synchronizes the write and read pointers is also converted by a second Gray code to sequential count converter 308.

Detailed Description Text (8):
For determining the level of the memory use in the FIFO, the write pointer and the read pointer have to be compared. A synchronization circuit 307 is used to synchronize the two pointers before they are compared by a subtractor 309 so that the comparison result can be stable and accurate. More specifically, the error of the subtraction result between the write pointer 304 and the read pointer 305 can be limited in a special range after the synchronization process. The size of the required memory buffer for handling the error is thus reduced, and the operation of the FIFO becomes more efficient.

Detailed Description Text (9):
In order to reduce the possible transient states that may occur in the write pointer 304 and the read pointer 305, the synchronizing to read clock circuit 307 of this invention synchronizes only the write pointer 304 to the read clock RCLK. The read pointer 305 is not synchronized. Therefore, transient states only occur when the write pointer is sampled by the read clock RCLK. The synchronizing to read clock circuit 307 of FIG. 3 is shown in FIG. 5. The circuit is implemented by a D-flip-flop. The write pointer is clocked through the flip-flop 501 synchronously with read clock RCLK to produce the output, synchronized write pointer.

Detailed Description Text (10):
For such a synchronization circuit as shown in FIG. 5, when the read clock RCLK synchronizes the write pointer 304, it may occur that the transient state from one address to its next address is sampled as illustrated in FIG. 6. As an example, in transferring data between AGP and SGRAM, the clock frequency is 66 MHz for AGP and 100 MHz for SGRAM. Therefore, the read frequency is faster than the write frequency. The write pointer is synchronized to the read clock of 100 MHz. A transient state in the write pointer may be sampled.

Detailed Description Text (11):
In this example the possible error after sampling is very small. The minimum error between the true value and the sampled result is one. The generated error is due to the fact that the address contained in an unstable transient state of the write pointer is sampled. The minimum error is one because in a Gray code counter no more than one bit is changed between two adjacent codes.

Detailed Description Text (12):
As mentioned earlier, the output of the synchronization circuit 307 is converted to a sequential count by a Gray code to sequential count converter 308. The synchronized and converted write pointer and the read pointer are then sent to the subtractor 309 in sequential count format for comparison. In general case, the bit size for the write pointer is equal to that for the read pointer. For example, if a FIFO having a memory size 32*64 is used, then the bit size for both write pointer and read pointer is 5 bits, and a 5-bit subtractor can meet the requirement of the design.

Detailed Description Text (19):
(2) The synchronizing to read clock circuit 307 is implemented by a D flip-flop, and it may delay a clock cycle (referred to as T.sub.D1) while a write pointer is being sampled by the read clock.

Detailed Description Text (21):
(4) It is possible that the output of the subtractor 309 indicates there is space for seven data units in the FIFO but the true available space can only hold six

data units due to the error caused by the transient state of the write pointer. Under the circumstance, if the AGP continues to write a block of data into the FIFO, the output from the subtractor 309 can be zero if the delays caused by the conditions in (2) and (3) also occur. In such a case, the status indicator would determine that the current status of the FIFO is empty. However, the true status of the FIFO is full. To prevent the wrong decision from being made, the invention adds one more data unit in the criteria (the extra clock cycle is referred to as T.sub.2).

Detailed Description Text (26):
As can be seen, while designing an asynchronous FIFO according to the preferred embodiment, it is necessary to precisely define an EMPTY signal in order to read all the data written into the FIFO. However, an extra buffer in the FIFO may be reserved for defining an ALMOST FULL signal to prevent an error from occurring. Therefore, the present invention uses the synchronizing to read clock circuit 307, as shown in FIG. 5, in which the synchronization is with respect to the read clock. It simplifies the determination of the EMPTY signal because a transient state only occurs in the write pointer but not in the read pointer. Therefore, in accordance with the invention, the method of determining an empty status of an asynchronous FIFO is unaffected by changes in the read or write frequency of the FIFO.

Detailed Description Text (29):
It is also worth mentioning that the write frequency in a general asynchronous FIFO may be faster than the read frequency in some case. Under the circumstance, an additional clock cycle T.sub.3 =1 has to be taken into account because a slower read clock is used to sample the write pointer. Two data units may have been written into the FIFO before a decision of the ALMOST FULL signal has been made. Consequently, the overall clock cycles for determining the issuing of the ALMOST FULL signal is T.sub.1 +T.sub.D1 +T.sub.D2 +T.sub.2 +T.sub.3 <=5. In other words, it is necessary for the status indicator 700 to issue an ALMOST FULL signal if the available space in the FIFO is equal to or less than five data units.

Detailed Description Text (30):
FIGS. 8a-8c illustrate how the write pointer may be sampled by the read clock when the write frequency is faster than the read frequency. In FIG. 8a, the write pointer changes only once between the two read clocks. FIG. 8b, however, shows that the write pointer changes twice between the two read clocks. In FIG. 8c, the first read clock occurs at the transient state of the write pointer and the sensed write pointer change may be one or two. Therefore, an additional data unit has to be included in determining the ALMOST FULL signal for the asynchronous FIFO.

Detailed Description Text (32):
In the alternative embodiment, the synchronization of the write printer and the read pointer is with respect to the write clock to simplify the determination of the FULL signal. The read pointer is implemented by a Gray code counter and the write pointer is implemented by a sequential counter. Therefore, two Gray code to sequential count converters are used to convert the address indicated by the read pointer from Gray code format to sequential count format.

Detailed Description Text (33):
In this embodiment, a transient state only occurs in the read pointer. Extra buffer space in the FIFO is reserved for defining a NEAR EMPTY signal to prevent an error from occurring. The principle for designing the status indicator in the embodiment of FIG. 3 can be applied to the design of the status indicator of FIG. 9. Because of the similarity between FIG. 3 and FIG. 9, the detail of the circuit for the asynchronous dual port FIFO of this embodiment is not described here.

CLAIMS:

1. An asynchronous dual port first-in-first-out memory buffer comprising:

a dual-port random access memory;

a read pointer indicating a read address for reading data words from said random access memory in response to a read clock, said read address being in a first format;

a write pointer indicating a write address in a second format;

a first converter for converting said write address from said second format to said first format for writing data words into said random access memory in response to a write clock;

a synchronization circuit for synchronizing said write pointer to said read clock so as to output a synchronized write pointer for indicating a synchronized write address in said second format;

a second converter for converting said synchronized write address from said second format to said first format;

a subtractor for obtaining the difference between said synchronized and converted write address and said read address; and

a status indicator receiving said difference and generating status signals for indicating the use level of said first-in-first-out memory buffer;

wherein said status signals comprise a first status signal for indicating that said first-in-first-out memory buffer is almost full when the 2's complement of said difference from said subtractor is not greater than a pre-determined critical value.

2. The asynchronous dual port first-in-first-out memory buffer according to claim 1, wherein said synchronization circuit synchronizes said write pointer to said read pointer using said read clock.

3. The asynchronous dual port first-in-first-out memory buffer according to claim 1, wherein said write pointer is a Gray code counter, said first and second converters are Gray code to sequential count converters and said read pointer is a sequential counter.

7. A method of controlling the access of an asynchronous dual port first-in-first-out memory buffer having a dual-port random access memory, comprising the steps of:

providing a read pointer containing a read address in a first format for reading data words from said random access memory in response to a read clock having a first frequency f.sub.1 ;

providing a write pointer containing a write address in a second format;

converting said write address from said second format to said first format for writing data words into said random access memory in response to a write clock having a second frequency f.sub.2 ;

providing a synchronization circuit for synchronizing said write pointer to said read pointer so as to output a synchronized write pointer containing a synchronized write address in said second format;

converting said synchronized write address in said second format to said first format;

computing the difference between said synchronized and converted write address and said read address in said first format; and

asserting a first status signal for indicating that said first-in-first-out memory buffer is almost full when the 2's complement of said difference is not greater than a pre-determined critical value.

8. The method according to claim 7, wherein synchronizing said write pointer to said read pointer is accomplished by synchronizing said write pointer to said read clock.

16. An asynchronous dual port first-in-first-out memory buffer comprising:

a dual-port random access memory;

a write pointer indicating a write address for writing data words into said random access memory in response to a write clock, said write address being in a first format;

a read pointer indicating a read address in a second format;

a first converter for converting said read address from said second format to said first format for reading data words from said random access memory in response to a read clock;

a synchronization circuit for synchronizing said read pointer to said write pointer so as to output a synchronized read pointer for indicating a synchronized read address in said second format;

a second converter for converting said synchronized read address from said second format to said first format;

a subtractor for obtaining the difference between said synchronized and converted read address and said write address; and

a status indicator receiving said difference and generating status signals for indicating the use level of said first-in-first-out memory buffer;

wherein said status signals comprise a first status signal for indicating that said first-in-first-out memory buffer is near empty when the 2's complement of said difference from said subtractor is not greater than a pre-determined critical value.

17. The asynchronous dual port first-in-first-out memory buffer according to claim 16, wherein said synchronization circuit synchronizes said read pointer to said write pointer using said write clock.

18. The asynchronous dual port first-in-first-out memory buffer according to claim 16, wherein said read pointer is a Gray code counter, said first and second converters are Gray code to sequential count converters and said write pointer is a sequential counter.